Efficient Tracking of Large Classes of Experts

András György, Tamás Linder, and Gábor Lugosi

October 12, 2011

Abstract

In the framework for prediction of individual sequences, sequential prediction methods are to be constructed that perform nearly as well as the best expert from a given class. We consider prediction strategies that compete with the class of switching strategies that can segment a given sequence into several blocks, and follow the advice of a different "base" expert in each block. As usual, the performance of the algorithm is measured by the regret defined as the excess loss relative to the best switching strategy selected in hindsight for the particular sequence to be predicted. In this paper we construct prediction strategies of low computational cost for the case where the set of base experts is large. In particular we derive a family of efficient tracking algorithms that, for any prediction algorithm $\mathcal{A}$ designed for the base class, can be implemented with time and space complexity $O(n^\gamma \log n)$ times larger than that of $\mathcal{A}$, where n is the time horizon and $\gamma \geq 0$ is a parameter of the algorithm. With $\mathcal{A}$ properly chosen, our algorithm achieves a regret bound of optimal order for $\gamma > 0$, and only $O(\log n)$ times larger than the optimal order for $\gamma = 0$ for all typical regret bound types we examined. For example, for predicting binary sequences with switching parameters, our method achieves the optimal $O(\log n)$ regret rate with time complexity $O(n^{1+\gamma} \log n)$ for any $\gamma \in (0, 1)$.

I. Introduction

A. Prediction with expert advice

In the on-line (sequential) decision problems considered in this paper, a decision maker (or forecaster) chooses, at each time instance $t = 1, 2, \ldots,$ an action from a set $\mathcal{D}$. After each action taken, the decision maker suffers some loss based on the state of the environment and the chosen decision. The general goal of the forecaster is to minimize its cumulative loss. Specifically, the forecaster's aim is to achieve a cumulative loss that is not much larger than that of the best expert (forecaster) from a reference class $\mathcal{E}$, where the best expert is chosen in hindsight. This problem is known as "prediction with expert advice." We refer to [1] for a survey.

Formally, let the decision space $\mathcal{D}$ be a convex subset of a vector space and let $\mathcal{Y}$ be a set representing the outcome space. Let $\ell : \mathcal{D} \times \mathcal{Y} \to \mathbb{R}$ be a loss function, assumed to be convex in its first argument.

This research was supported in part by the Hungarian Scientific Research Fund and the Hungarian National Office for Research and Technology (OTKA F60787, OTKA-NKTH CNK 77782), the Natural Sciences and Engineering Research Council (NSERC) of Canada, the Spanish Ministry of Science and Technology grant MTM2009-09063 and the PASCAL2 Network of Excellence under EC grant no. 216886.

A. György is with the Machine Learning Research Group, Computer and Automation Research Institute of the Hungarian Academy of Sciences, Kende u. 13-17, Budapest, Hungary, H-1111 (email: gya@szit.bme.hu). T. Linder is with the Department of Mathematics and Statistics, Queen's University, Kingston, Ontario, Canada K7L 3N6 (email: linder@mast.queensu.ca). G. Lugosi is with ICREA and the Department of Economics, Pompeu Fabra University, Ramon Trias Fargas 25-27, 08005 Barcelona, Spain (email: gabor.lugosi@gmail.com).

PREDICTION WITH EXPERT ADVICE

For each round $t = 1, 2, \ldots$

(1) the environment chooses the next outcome y_t and the expert advice $\{f_{i,t} \in \mathcal{D} : i \in \mathcal{E}\}$; the expert advice is revealed to the forecaster;
(2) the forecaster chooses the prediction $\widehat{p}_t \in \mathcal{D}$;
(3) the environment reveals the next outcome $y_t \in \mathcal{Y}$;
(4) the forecaster incurs loss $\ell(\widehat{p}_t, y_t)$ and each expert i incurs loss $\ell(f_{i,t}, y_t)$.

Fig. 1. The repeated game of prediction with expert advice.

At each time instant $t = 1, \ldots, n$, the environment chooses an action $y_t \in \mathcal{Y}$ and each expert $i \in \mathcal{E}$ forms its prediction $f_{i,t} \in \mathcal{D}$. Then the forecaster chooses an action $\widehat{p}_t \in \mathcal{D}$ (without knowing y_t), suffers loss $\ell(\widehat{p}_t, y_t)$, and the losses $\ell(f_{i,t}, y_t), i \in \mathcal{E}$ are revealed to the forecaster. (This is known as the full information case and in this paper we only consider this model. In other, well-studied, variants of the problem, the forecaster only receives limited information about the outcome.)

The goal of the forecaster is to minimize its cumulative loss $\widehat{L}_n = \sum_{t=1}^n \ell(\widehat{p}_t, y_t)$, which is equivalent to minimizing its excess loss $\widehat{L}_n - \min_{i \in \mathcal{E}} L_{i,n}$ relative to the the set of experts $\mathcal{E}$, where $L_{i,n} = \sum_{t=1}^n \ell(f_{i,t}, y_t)$ for all $i \in \mathcal{E}$.

Several methods are known that can compete successfully with different expert classes $\mathcal{E}$ in the sense that the (worst-case) cumulative regret, defined as

$$R_n = \max_{y_1,\ldots,y_n \in \mathcal{Y}^n} \left(\widehat{L}_n - \min_{i \in \mathcal{E}} L_{i,n} \right) = \max_{y_1,\ldots,y_n \in \mathcal{Y}^n} \left(\sum_{t=1}^n \ell(\widehat{p}_t, y_t) - \min_{i \in \mathcal{E}} \sum_{t=1}^n \ell(f_{i,t}, y_t) \right)$$

only grows sub-linearly, that is, $\lim_{n \to \infty} R_n/n = 0$. One of the most popular among these is *exponential weighting*. When the expert class $\mathcal{E}$ is finite of countably infinite, this method assigns, at each time instant t, a nonnegative weight

$$\pi_{i,t} = \frac{w_i e^{-\eta_t L_{i,t-1}}}{\sum_{j \in \mathcal{E}} w_j e^{-\eta_t L_{i,t-1}}}$$

to each expert $i \in \mathcal{E}$. Here $L_{i,t-1} = \sum_{\tau=1}^{t-1} \ell(f_{i,\tau}, y_\tau)$ is the cumulative loss of expert i up to time $t-1$, $\eta_t > 0$ is some learning parameter, and the $w_i > 0$ are nonnegative initial weights with $\sum_{i \in \mathcal{E}} w_i = 1$, so that $\sum_{i \in \mathcal{E}} \pi_{i,t} = 1$ (we define $L_{i,0} = 0$ for all $i \in \mathcal{E}$, as well as $\widehat{L}_0 = 0$). The decision chosen by this algorithm is

$$\widehat{p}_t = \sum_{i \in \mathcal{E}} \pi_{i,t} f_{i,t}. \tag{1}$$

In this paper we concentrate on two special types of loss functions: bounded convex and exp-concave. For such loss functions the regret of the exponentially weighted average forecaster is well understood. For example, assume ℓ is convex in its first argument and takes its values in $[0,1]$, and the set of experts is finite with $|\mathcal{E}| = N$. Then if η_t is nonincreasing in t then, for any n,

$$\widehat{L}_n \le \min_i \left\{ L_{i,n} + \frac{1}{\eta_n} \ln \frac{1}{w_i} \right\} + \sum_{t=1}^{n} \frac{\eta_t}{8} \tag{2}$$

see [2]. By setting the initial weights to $w_i = 1/N, i = 1, \dots, N$ and with the choice $\eta_t = 2\sqrt{\ln N/t}$, one obtains, for any $n \ge 1$,

$$R_n \le \sqrt{n \ln N}\ . \tag{3}$$

If, on the other hand, for some $\eta > 0$ the function $F(p) = e^{-\eta \ell(p,y)}$ is concave for any fixed $y \in \mathcal{Y}$ (such loss functions are called *exp-concave*) then, choosing $\eta_t \equiv \eta$ and $w_i = 1/N, i = 1, \dots, N$, one has for any $n \ge 1$,

$$R_n \le \frac{\ln N}{\eta}\ . \tag{4}$$

The family of exp-concave loss functions includes, for example, for $p, y \in [0,1]$, the square loss $\ell(p,y) = (p-y)^2$ with $\eta \le 1/2$, and the relative entropy loss $\ell(p,y) = y \ln \frac{y}{p} + (1-y) \ln \frac{1-y}{1-p}$ with $\eta \le 1$. A special case of the latter is the logarithmic loss defined for $y \in \{0,1\}$ and $p \in [0,1]$ by $\ell(p,y) = -\mathbb{I}_{y=1} \log p - \mathbb{I}_{y=0} \log(1-p)$, which plays a central role in data compression. Here and throughout the paper $\mathbb{I}_B$ denotes the indicator of event B and all logarithms are to the base 2. We refer to [1] for discussion of these bounds.

A naive implementation of the exponentially weighted average forecaster maintains one weight for each of the N experts, and the algorithm can be performed using $O(N)$ operations per time round. However, when N is large or infinite, more efficient methods are called for. There are many examples of finite or infinite expert classes for which efficient algorithms yielding good regret bounds are known. These algorithms utilize the structure inherent in the problem that allows a more compact representation leading to low complexity solutions. Examples include, among others, the logarithmic loss with the set of experts consisting of all probability distributions on a finite set (here the well-known Krichevsky-Trofimov method yields an efficient predictor [3]), or the class of all bounded-memory Markov sources (see the context tree weighting method of [4]); the case when the set of experts consists of paths in a directed graph such that the loss of a path is obtained as the sum of the losses corresponding to its edges [5]; the problem of predicting as well as the best linear predictor [6], [7], [8]; or the problem of predicting as well as the best convex combination of a set of base experts [9].

B. *Tracking the best expert*

The goal of the standard online prediction problem described in the previous section is to perform nearly as well as the best expert in the class $\mathcal{E}$. A more ambitious goal is to compete with the best *sequence* of expert predictions that may switch its experts at a certain, limited, number of times. This, seemingly more complex, problem may be regarded as a special case of the standard setup by introducing the so-called *meta experts*. A meta expert is described by a sequence of base experts $(i_1, \dots, i_n) \in \mathcal{E}^n$, such that at time instants $t = 1, \dots, n$ the meta expert follows the prediction of the "base" expert $i_t \in \mathcal{E}$ by predicting $f_{i_t,t}$. The complexity of such a meta expert may be measured by $C = |\{t \in \{1, 2, \dots, n-1\} : i_t \ne i_{t+1}\}|$, the number of times it changes the base predictor (each such change is called a switch). Note that C switches partition $\{1,\ldots,n\}$ into $C+1$ contiguous segments, on each of which the meta expert's prediction is constant. If a maximum of m changes are allowed and the set of base experts has N elements, then the class of meta experts is of size $\sum_{j=0}^{m}\binom{n-1}{j}N(N-1)^j$. Clearly, a naive implementation of the exponentially weighted average forecaster is not feasible in this case, but several more efficient algorithms have been proposed.

One approach, widely used in the information theory/source coding literature, is based on transition diagrams [10], [11]: A transition diagram is used to define a prior distribution on the switches of the experts, and the starting point of the current segment is estimated using this prior. In its straightforward version, at each time instant t, the performance of an expert algorithm is emulated for all possible segment starting points $1,\ldots,t$, and a weighted average of the resulting estimates is used to form the next prediction. In effect, this method converts an efficient algorithm to compete with the best expert in a class $\mathcal{E}$ into one that competes with the best sequence of experts with a limited number of changes. However, the time complexity of the resulting algorithm increases by a factor of n, the time horizon compared with the original algorithm that competes with $\mathcal{E}$, yielding a total complexity that is quadratic in n.

For the same problem, a method of linear complexity was developed in [12], but it requires an a priori known upper bound on the number of switches, while transition-diagram based methods can adapt to an arbitrary number of switches. (Of course, the regret scales with the number of switches.) Vovk [13] showed that the method of [12] is equivalent to a an easy-to-implement weighting of the paths in the full transition diagram. The algorithm can be modified to compete with meta experts with an arbitrary number of switches: a linear complexity variant achieves this goal (by letting its switching parameter α decrease to zero) at the price of somewhat increasing the regret [14]. A slightly better regret bound can be achieved for the case when switching occurs more often at the price of increasing the computational complexity from linear to $O(n^3/2)$ [15], [16] (by discretizing its switching parameter α to $\sqrt{n}$ levels). On the other hand, reduced transition diagrams have been used for the logarithmic loss (i.e., data compression) by [17] and by [11] (the latter work considers a probabilistic setup as opposed to the individual sequence setting). An efficient algorithm based on a reduced transition diagram for the general tracking problem was given in [18] , while [19] developed independently a similar algorithm to minimize the adaptive regret

$$R_n^a = \max_{t\le t'} \max_{y_t,y_{t+1},\ldots,y_{t'}} \left(\sum_{\tau=t}^{t'} \ell(\hat{p}_\tau, y_\tau) - \min_{i\in\mathcal{E}} \ell(f_{i,\tau}, y_\tau) \right)$$

which is the maximal worst-case cumulative excess loss over any contiguous time segment relative to a constant expert. It is clear that the regret of an algorithm, in n time steps, relative to a meta expert that can switch the base expert C times can be bounded by $(C+1)R_n^a$, so the bounds in [19] also provide bounds for the tracking problem.

An important question is how one can compete with meta experts when the base expert class $\mathcal{E}$ is very large. In such cases special algorithms are needed to compete with experts from the base class even without switching. Such large base classes arise in on-line linear optimization [9], lossless data compression [3], [4], the shortest path problem [5], [20], or limited-delay lossy data compression [21]–[23]. Such special algorithms can easily be incorporated in transition-diagram-based tracking methods, but the resulting complexity is quadratic in n (see, e.g., [11] for such an application to lossless data compression or [24]–[26] for applications to signal processing and universal portfolio selection). If the special algorithms for large base expert classes are combined with the algorithm of [12] to compete with meta experts, the resulting algorithms again have quadratic complexity in n; see, e.g., [13], [27].

In this paper we tackle the complexity issue by presenting a general method for designing reduced transition diagrams. Our algorithm unifies and generalizes the algorithms of [17], [19] and our earlier work [18]. This algorithm has an explicit complexity-regret trade-off, covering essentially all such results in the literature. In addition to the (almost) linear complexity algorithms in the aforementioned papers, the parameters of our algorithm can be set to reproduce the methods based on the full transition diagram [10], [11], [24], or the complexity-regret behavior of [15], [16]. Also, our algorithm has regret of optimal order with complexity $O(n^{1+\gamma}\log n)$ for any $\gamma \in (0,1)$, while setting $\gamma = 0$ results in complexity $O(n \log n)$ and a regret that is only a factor of $\log n$ larger than the optimal rate (similarly to [17]–[19]).

The rest of the paper is organized as follows. In Section II-A we describe our general algorithm. Sections II-B and II-C present a unified method for the low-complexity implementation of the general algorithm via reduced transition diagrams. Bounds on the algorithm are developed in Section II-D. More explicit bounds are presented for some important special cases in Sections II-E and II-F. The results are extended to the related framework of randomized prediction in Section III. Some applications to concrete examples are given in Section IV.

II. A REDUCED COMPLEXITY TRACKING ALGORITHM

A. A general tracking algorithm

Here we introduce a general tracking method which forms the basis of or reduced complexity tracking algorithm. Consider an on-line forecasting algorithm $\mathcal{A}$ that chooses an element of the decision space depending on the past outcomes and the expert advices according to the protocol described in Figure 1. Suppose that for all n and possible outcome sequences of length n, $\mathcal{A}$ satisfies a regret bound

$$R_n \le \rho_{\mathcal{E}}(n) \tag{5}$$

with respect to the base expert class $\mathcal{E}$, where $\rho_{\mathcal{E}} : [0,\infty) \to [0,\infty)$ is a nondecreasing and concave function with $\rho_{\mathcal{E}}(0) = 0$. These assumptions on $\rho_{\mathcal{E}}$ are usually satisfied by the known regret bounds for different algorithms, such as the bounds (3) and (4) (with defining $\rho_{\mathcal{E}}(0) = 0$ in the latter case). Suppose $1 \le t_1 < t_2 \le n$ and an instance of $\mathcal{A}$ is used for time instants $t \in [t_1, t_2) := \{t_1, \ldots, t_2 - 1\}$, that is, algorithm $\mathcal{A}$ is run on data obtained in the segment $[t_1, t_2)$. The accumulated loss of $\mathcal{A}$ during this period will be denoted by $L_{\mathcal{A}}(t_1, t_2)$. Then (5) implies

$$L_{\mathcal{A}}(t_1, t_2) - \min_{i \in \mathcal{E}} L_i(t_1, t_2) \le \rho_{\mathcal{E}}(t_2 - t_1)$$

where $L_i(t_1, t_2) = \sum_{t=t_1}^{t_2-1} \ell(f_{i,t}, y_t)$ denotes the loss of expert i in the interval $[t_1, t_2)$.

Fix the time horizon $n \ge 1$. A meta expert that changes base experts at most $C \ge 0$ times can be described by a vector of experts $a = (i_0, \ldots, i_C) \in \mathcal{E}^{C+1}$ and a "transition path" $T = (t_1, \ldots, t_C; n)$ such that $t_0 := 1 < t_1 < \ldots < t_C < t_{C+1} := n + 1$. For each $c = 0, \ldots, C$, the meta expert follows the advice of expert i_c in the time interval $[t_c, t_{c+1})$. When the time horizon n is clear from the context, we will omit it from the description of T and simply write $T = (t_1, \ldots, t_C)$. We note that this representation is not unique as the definition does not require that base experts i_c and i_{c+1} be different. Any meta expert that can be defined using a given transition path T is said to follow T.

The total loss of the meta expert indexed by (T, a), accumulated during n rounds, is

$$L_n(T, a) = \sum_{c=0}^{C} L_{i_c}(t_c, t_{c+1}) \ .$$

For any $t \geq 1$, let $\mathcal{T}_t$ denote the set of all transition paths up to time t represented by vectors $(t_1, \ldots, t_C; t)$ with $1 < t_1 < t_2 < \ldots < t_C \leq t$ and $0 \leq C \leq t$. For any $T = (t_1, \ldots, t_C) \in \mathcal{T}_n$ and $t \leq n$ define the truncation of T at time t as $T_t = (t_1, \ldots, t_k; t)$, where k is such that $t_k \leq t < t_{k+1}$. Furthermore, let $\tau_t(T) = \tau_t(T_t) = t_k$ denote the last change up to time t, and let $C_t(T) = C(T_t) = k$ denote the number of switches up to time t. A transition path T with C switches splits the time interval $[1, n]$ into $C+1$ contiguous segments. We apply algorithm $\mathcal{A}$ on T in such a way that at the beginning of each segment (at time instants t_c) we restart $\mathcal{A}$; this algorithm will be denoted in the sequel by $(\mathcal{A}, T)$. Denote the output of the algorithm at time t by $f_{\mathcal{A},t}(T_t) = f_{\mathcal{A},t}(\tau_t(T))$. This notation emphasizes the fact that, since $\mathcal{A}$ is restarted at the beginning of each segment of T, its output at time t depends only on $\tau_t(T)$, the beginning of the segment which includes t. The loss of algorithm $(\mathcal{A}, T)$ up to time n is

$$L_n(\mathcal{A}, T) = \sum_{c=0}^{C} L_{\mathcal{A}}(t_c, t_{c+1}) \ .$$

As most tracking algorithms, our algorithm will use weight functions $w_t : \mathcal{T}_t \to [0, 1]$ satisfying

$$\sum_{T \in \mathcal{T}_t} w_t(T_t) = 1 \qquad \text{and} \qquad w_t(T_t) = \sum_{T'_{t+1} : T'_t = T_t} w_{t+1}(T'_{t+1}) \ . \tag{6}$$

Thus each w_t is a probability distribution on $\mathcal{T}_t$ such that the family $\{w_t; t = 1, \ldots, n\}$ is consistent. To simplify the notation, we formally define T_0 as the "empty transition path" $\mathcal{T}_0 := \{T_0\}$, $L_0(\mathcal{A}, T_0) := 0$, and $w_0(T_0) := 1$.

We say that $\widehat{T} \in \mathcal{T}_n$ *covers* $T \in \mathcal{T}_n$ if the change points of T are also change points of $\widehat{T}$. Note that if $\widehat{T}$ covers T, then any meta expert that follows transition path T also follows transition path $\widehat{T}$. We say that w_n *covers* $\mathcal{T}_n$ if for any $T \in \mathcal{T}_n$ there exists a $\widehat{T} \in \mathcal{T}_n$ with $w_n(\widehat{T}) > 0$ which covers T.

Now we are ready to define our first master algorithm, given in Algorithm 1.

Algorithm 1 General tracking algorithm.

```
Input: prediction algorithm $\mathcal{A}$, weight functions $\{w_t; t = 1, \ldots, n\}$, learning parameters $\eta_t > 0, t = 1, \ldots, n$.
For $t = 1, \ldots, n$ predict
    $\widehat{p}_t = \frac{\sum_{T \in \mathcal{T}_t} w_t(T) e^{-\eta_t L_{t-1}(\mathcal{A}, T_{t-1})} f_{\mathcal{A},t}(\tau_t(T))}{\sum_{T \in \mathcal{T}_t} w_t(T) e^{-\eta_t L_{t-1}(\mathcal{A}, T_{t-1})}}$ .
```

We note that the consistency of $\{w_t\}$ implies that, for any time horizon n, Algorithm 1 is equivalent to the exponentially weighted average forecaster (1) with set of experts $\{(\mathcal{A}, T) : T \in \mathcal{T}_n, w_n(T_n) > 0\}$ and initial weights $w_n(T)$ for $(\mathcal{A}, T)$.

The next lemma gives an upper bound on the performance of Algorithm 1.

Lemma 1: Suppose $\eta_{t+1} \le \eta_t$ for all $t = 1, \ldots, n-1$, the transition path T_n is covered by $\widehat{T}_n = (\hat{t}_1, \ldots, \hat{t}_{C(\widehat{T}_n)})$ such that $w_n(\widehat{T}_n) > 0$, and $\mathcal{A}$ satisfies the regret bound (5). Assume that the loss function ℓ is convex in its first argument and takes values in the interval $[0,1]$. Then for any meta expert (T_n, a) the regret of Algorithm 1 is bounded as

$$\begin{aligned}\widehat{L}_n - L_n(T_n, a) &\le \sum_{c=0}^{C(\widehat{T}_n)} \rho_{\mathcal{E}}(\hat{t}_{c+1} - \hat{t}_c) + \sum_{t=1}^{n} \frac{\eta_t}{8} + \frac{1}{\eta_n} \ln \frac{1}{w_n(\widehat{T}_n)} \\ &\le (C(\widehat{T}_n) + 1)\rho_{\mathcal{E}}\left(\frac{n}{C(\widehat{T}_n)+1}\right) + \sum_{t=1}^{n} \frac{\eta_t}{8} + \frac{1}{\eta_n} \ln \frac{1}{w_n(\widehat{T}_n)} \ .\end{aligned} \tag{7}$$

On the other hand, if ℓ is exp-concave for the value of η and Algorithm 1 is used with $\eta_t \equiv \eta$, then

$$\begin{aligned}\widehat{L}_n - L_n(T_n, a) &\le \sum_{c=0}^{C(\widehat{T}_n)} \rho_{\mathcal{E}}(\hat{t}_{c+1} - \hat{t}_c) + \frac{1}{\eta} \ln \frac{1}{w_n(\widehat{T}_n)} \\ &\le (C(\widehat{T}_n) + 1)\rho_{\mathcal{E}}\left(\frac{n}{C(\widehat{T}_n)+1}\right) + \frac{1}{\eta} \ln \frac{1}{w_n(\widehat{T}_n)} \ .\end{aligned} \tag{8}$$

Proof: Let $\hat{a} = (\hat{\imath}_0, \ldots, \hat{\imath}_C)$ be the expert vector such that the meta experts (T, a) and $(\widehat{T}, \hat{a})$ perform identically. Then clearly

$$\widehat{L}_n - L_n(T, a) = \widehat{L}_n - L_n(\mathcal{A}, \widehat{T}_n) + L_n(\mathcal{A}, \widehat{T}_n) - L_n(\widehat{T}_n, \hat{a}) \ .$$

Using (5) and the concavity of $\rho_{\mathcal{E}}$, we get

$$\begin{aligned}L_n(\mathcal{A}, \widehat{T}_n) - L_n(\widehat{T}_n, \hat{a}) &= \sum_{c=0}^{C(\widehat{T}_n)} \Big(L_{\mathcal{A}}(\hat{t}_c, \hat{t}_{c+1}) - L_{\hat{\imath}_c}(\hat{t}_c, \hat{t}_{c+1}) \Big) \\ &\le \sum_{c=0}^{C(\widehat{T}_n)} \rho_{\mathcal{E}}(\hat{t}_{c+1} - \hat{t}_c) \le (C(\widehat{T}_n) + 1)\rho_{\mathcal{E}}\left(\frac{n}{C(\widehat{T}_n)+1}\right) \ .\end{aligned} \tag{9}$$

Assume that the loss function ℓ is convex in its first argument and takes values in the interval $[0,1]$. Since Algorithm 1 is equivalent to the exponentially weighted average forecaster with experts $\{(\mathcal{A}, T) : T \in \mathcal{T}_n, w_n(T) > 0\}$ and initial weights $w_n(T)$ we can apply the bound (2) to obtain

$$\widehat{L}_n \le L_n(\mathcal{A}, \widehat{T}_n) + \frac{1}{\eta} \ln \frac{1}{w_n(\widehat{T}_n)} + \sum_{t=1}^{n} \frac{\eta_t}{8}.$$

Combining this with (9) proves (7).

Now assume ℓ is exp-concave. Then by [12, Lemma 1],

$$\widehat{L}_n - L_n(\mathcal{A}, \widehat{T}_n) \le \frac{1}{\eta} \ln \frac{1}{w_n(\widehat{T}_n)} \ . \tag{10}$$

This, together with (9), implies (8).

■

B. The weight function

One may interpret the weight function $\{w_t\}$ as the conditional probability that a new segment is started, given the beginning of the current segment and the current time instant. In this case one may define $\{w_t\}$ in terms of a time-inhomogeneous Markov chain $\{U_t;\, t = 1, 2, \ldots\}$ whose state space at time t is $\{1, \ldots, t\}$. The distribution of $\{U_t\}$ is uniquely determined by prescribing $\mathbb{P}(U_1 = 1) = 1$ and for $1 \le t' < t$,

$$\mathbb{P}(U_t = t | U_{t-1} = t') = 1 - \mathbb{P}(U_t = t' | U_{t-1} = t') = p(t|t') \tag{11}$$

where the so-called *switch probabilities* $p(t|t')$ need only satisfy $p(t|t') \in [0, 1]$ for all $1 \le t' < t$. (Thus, this Markov chain, at time t, either stays where it was at time $t - 1$ or jumps to state t.) A realization of this Markov chain uniquely determines a transition path: $T_t(u_1, \ldots, u_t) = (t_1, \ldots, t_C) \in \mathcal{T}_t$ if and only if $u_{k-1} \ne u_k$ for $k \in \{t_1, \ldots, t_C\}$, and $u_{k-1} = u_k$ for $k \notin \{t_1, \ldots, t_C\}$, $2 \le k \le t$. Inverting this correspondence, any $T \in \mathcal{T}_t$ uniquely determines a realization $(u_1, \ldots, u_t)$. Now the weight function is given for all $t \ge 1$ and $T \in \mathcal{T}_t$ by

$$w_t(T) = \mathbb{P}(U_1 = u_1, \ldots, U_t = u_t) \tag{12}$$

where $(u_1, \ldots, u_t)$ is such that $T = T(u_1, \ldots, u_t)$. It is easy to check that $\{w_t\}$ satisfies the two conditions in (6). Clearly, the switch probabilities $p(t|t')$ uniquely determine $\{w_t\}$.

Some examples that have been proposed for this construction (given in terms of the switch probabilities) include

- w^{HW}, used in [12], is defined by $p_{HW}(t|t') = \alpha$ for some $0 < \alpha < 1$.
- w^{HS}, used in [14], [16], [19], is defined by $p^{HS}(t|t') = 1/t$.
- w^{KT}, used in [10], is defined by

$$p_{KT}(t|t') = \frac{1/2}{t - t' + 1}$$

which is the Krichevsky-Trofimov estimate [3] for binary sequences of the probability that after observing an all zero sequence of length $t - t'$, the next symbol will be a one. Using standard bounds on the Krichevsky-Trofimov estimate, it is easy to show (see, e.g., [10]) that for any $T \in \mathcal{T}_n$ with segment lengths $s_0, s_1, \ldots, s_C \ge 1$ (satisfying $\sum_{c=0}^{C} s_c = n$)

$$\ln \frac{1}{w^{KT}(T)} \le \frac{1}{2} \sum_{c=0}^{C} \ln s_c + (C + 1) \ln 2. \tag{13}$$

- $w^{\mathcal{L}_1}$ and $w^{\mathcal{L}_2}$ used in [11] (similar weight functions were considered in [13]), are defined as follows: for a given $\epsilon > 0$, let $\pi_j = 1/j^{1+\epsilon}$, $Z_t = \sum_{j=1}^{t} \pi(j)$ and $Z_\infty = \sum_{j=1}^{\infty} \pi(j)$. Then $w^{\mathcal{L}_1}$ and $w^{\mathcal{L}_2}$ are defined, respectively, by

$$p_{\mathcal{L}_1}(t|t') = \frac{\pi(t-1)}{(Z_\infty - Z_{t-2})} \quad \text{and} \quad p_{\mathcal{L}_2}(t|t') = \frac{\pi(t - t')}{(Z_\infty - Z_{t-t'+1})}.$$

Here we consider the weights $w^{\mathcal{L}_1}$. It is shown in [11, proof of Eq. (39)] that for any $T \in \mathcal{T}_n$,

$$\ln \frac{1}{w_n^{\mathcal{L}_1}(T)} \le (C_n(T) + \epsilon) \ln n + \ln(1 + \epsilon) - C_n(T) \ln \epsilon\ . \tag{14}$$

C. A low-complexity algorithm

Efficient implementation of Algorithm 1 hinges on three factors: (i) Algorithm $\mathcal{A}$ can be efficiently implemented; (ii) the exponential weighting step can be efficiently implemented; which is facilitated by (iii) the availability of the losses $L_{\mathcal{A},t}$. In what follows we assume that (i) and (iii) hold and develop a method for (ii) via constructing a new weight function $\{\hat{w}_t\}$ that significantly reduces the complexity of implementing Algorithm 1.

First, we observe that the predictor $\hat{p}_t$ of Algorithm 1 can be rewritten as

$$\widehat{p}_t = \frac{\sum_{t'=1}^{t} v_t(t') e^{-\eta_t L_{\mathcal{A}}(t',t-1)} f_{\mathcal{A},t}(t')}{\sum_{t'=1}^{t} v_t(t') e^{-\eta_t L_{\mathcal{A}}(t',t-1)}} \tag{15}$$

where the weights v_t are given by

$$v_t(t') = \sum_{T \in \mathcal{T}_t \colon \tau_t(T)=t'} w_t(T) e^{-\eta_t L_{t'-1}(\mathcal{A}, T_{t'-1})}. \tag{16}$$

If the learning parameters η_t are constant during the time horizon, the above means that Algorithm 1 can be implemented efficiently by keeping a weight $v_t(t')$ at each time instant t for every possible starting point of a segment $t' = 1, \ldots, t$. Indeed, if $\eta_t = \eta$ for all t, then (16), (11), and (12) imply that each $v_t(t')$ can be computed recursively in $O(t)$ time from the v_{t-1} (setting $v_1(1) := 1$ at the beginning) using the switch probabilities defining w_t as follows:

$$v_t(t') = \begin{cases} v_{t-1}(t')(1 - p(t|t')) e^{-\eta \ell(f_{\mathcal{A},t-1}(t'), y_{t-1})} & \text{for } t' = 1, \ldots, t-1, \\ \sum_{t'=1}^{t-1} v_{t-1}(t') p(t|t') e^{-\eta \ell(f_{\mathcal{A},t-1}(t'), y_{t-1})} & \text{for } t' = t. \end{cases} \tag{17}$$

Using this recursion, the overall complexity of computing the weights during n rounds is $O(n^2)$. Furthermore, (15) means that one needs to start an instance of $\mathcal{A}$ for each possible starting point of a segment. If the complexity of running algorithm $\mathcal{A}$ for n time steps is $O(n)$ (i.e., computing $\mathcal{A}$ at each time instance has complexity $O(1)$), then the overall complexity of our algorithm becomes $O(n^2)$.

It is clearly not a desirable feature that the amount of computation per time round grows (linearly) with the horizon n. While we don't know how to completely eliminate this ever-growing computational demand, we are able to moderate this growth significantly. To this end, we modify the weight functions in such a way that at any time instant t we allow at most $O(g \log t)$ actual segments with positive probability (i.e., segments containing t that belong to sample paths with positive weights), where $g > 0$ is a parameter of the algorithm (note that g may depend on, e.g., the time horizon n). Specifically, we will construct a new weight function $\hat{w}_t$ such that

$$\left|\{\tau_t(T) : \hat{w}_t(T_t) > 0, T \in \mathcal{T}_n\}\right| \leq g \log t.$$

By doing so, the time and space complexity of the algorithm becomes $O(g \log n)$ times more than that of algorithm $\mathcal{A}$, as we need to run $O(g \log n)$ instances of $\mathcal{A}$ in parallel and the number of non-zero terms in (17) and (15) is also $O(g \log n)$. Thus, in case of a linear-time-complexity algorithm $\mathcal{A}$, the overall complexity of Algorithm 1 becomes $O(gn \log n)$. (To be precise, to achieve the space complexity $O(g \log n)$ times that of $\mathcal{A}$ we need that the space complexity of $\mathcal{A}$ be at least a positive constant as we need to store $O(g \log n)$ weights; note that the time complexity of $\mathcal{A}$ must be at least linear in n).

In order to construct the new weight function, at each time instant t we force some segments to end. Then any path that contains such a segment will start a new segment at time t (and hence the corresponding vector of transitions contains t). Specifically, if a segment starts at time instant s, where s can be written as $o2^u$ with o being an odd number and u an integer, $o, u \geq 0$ (that is, 2^u is the largest power of 2 that divides t), then s can "live" for at most $g2^u$ time instances, where $g > 0$ is a parameter of the algorithm. Thus at time $s + g2^u$ we force a switch in the path. More precisely, given any switching probability $p(t|t')$ for all $t' < t$, we define a new switching probability

$$\hat{p}(t|t') = 1 - h_t(t')\big(1 - p(t|t')\big) \tag{18}$$

where

$$h_t(s) = \begin{cases} 1 & \text{if } s \leq t < s + g2^u, \\ 0 & \text{otherwise.} \end{cases}$$

Thus $h_t(s) = 1$ if and only if a segment started at s is still valid at time t. In this way, given the switching probabilities $p(t|t')$ and the associated weight function $\{w_t\}$, we can define a new weight function $\{\hat{w}_t\}$ via the new switching probabilities $\hat{p}(t|t')$ and the procedure described in Section II-B. Note that the definition of $\{\hat{w}_t\}$ implies that for a transition path $T \in \mathcal{T}_t$ either

$$\hat{w}_t(T) = 0 \qquad \text{or} \qquad \hat{w}_t(T) \geq w_t(T) \ . \tag{19}$$

The above procedure is a common generalization of previous algorithms in the literature for pruning the transition paths. Specifically, $g = 1$ yields the procedure of [17], $g = 3$ yields our previous procedure [18], $g = 4$ yields the method of [19], while $g = n$ yields the original weighting $\{w_t\}$ without pruning. We will show that the time complexity of the method with a constant g (i.e., when g is independent of the time horizon n) is, in each time instant, at most $O(\log n)$ times the complexity of one step of $\mathcal{A}$, while the time complexity of the algorithm without pruning is $O(n)$ times the complexity of $\mathcal{A}$. Complexities that interpolate between these two extremes can be achieved by setting $g = O(n)$ appropriately.

We say that a segment at time instant t is *alive* if it contains t and is *valid* if there is a path T_t with $\hat{w}_t(T_t) > 0$ that contains exactly that segment. In what follows we assume that the original switching probabilities $p(t|t')$ associated with the w_t satisfy $p(t|t') \in (0, 1)$ for all $1 \leq t' < t$. (Note that the weight function examples introduced in Section II-B all satisfy this condition.) The condition implies that $w_t(T_t) > 0$ for all $T_t \in \mathcal{T}_t$. Furthermore, if $T_t = (t_1, \ldots, t_C) \in \mathcal{T}_t$ satisfies $t_{i+1} - t_i < g2^{u_{t_i}}$, $i = 1, \ldots, C$, where u_{t_i} is the largest power of 2 divisor of t_i, then from (18) we get $\hat{w}_t(T) > 0$.

The next lemma gives a characterization of when $h_s(t) = 1$, and, as a consequence, bounds the number of valid segments that are alive at t.

Lemma 2: Let $t = \sum_{i=1}^{m} 2^{u_i}$ be the binary form of t with $0 \leq u_1 < u_2 < \cdots < u_m = u$, and let $s_k = \sum_{i=k}^{m} 2^{u_i}$. Then $h_t(s) = 1$ if and only if either (i) $s = s_k - j2^{u_k+1}$ for some $1 \leq k \leq m$, $0 \leq j < g$ or (ii) $s = s_m - (2j-1)2^l$ with $0 \leq l < u$, $l \neq u_i, i = 1, \ldots, m-1$, and $1 \leq j \leq g/2$. As a consequence, at any time instant t there are at most $g \log t$ segments that are valid and alive.

Proof: The proof is a direct consequence of the definition of $h_s(t)$, since any possible starting point s of a valid segment with largest 2-power divisor 2^l satisfies (i) if $l = u_i$ for some $i = 1, \ldots, m$, and satisfies (ii) if $l \neq u_i, i = 1, \ldots, m$. ■

Note that for $g = 1$ the valid segments that are alive at t start exactly at $s_k, k = 1, \ldots, m$, and so the number of valid segments at time t is exactly the number of 1's in the binary form of t [17]. The above lemma implies that Algorithm 1 can be implemented efficiently with the proposed weight function $\{\hat{w}_t\}$.

Theorem 1: Assume Algorithm 1 is run with weight function $\{\hat{w}_t\}$ derived using any $g > 0$ from any weight function $\{w_t\}$ defined as in Section II-B. If $\eta_t = \eta$ for some $\eta > 0$ and all $t = 1, \ldots, n$, then the time an space complexity of Algorithm 1 is $O(g \log n)$ times the time and space complexity of $\mathcal{A}$, respectively.

Proof: The result follows since Lemma 2 implies that the number of non-zero terms in (17) and (15) is always $O(g \log t)$. ■

D. Regret bounds

To bound the regret, we need the following lemma which shows that any segment $[t, t')$ can be covered with at most $\left\lceil \frac{\log(t'-t)}{\lfloor \log(g+1) \rfloor} \right\rceil + 1$ valid segments.

Lemma 3: For any $T \in T_n$, there exists $\widehat{T} \in T_n$ such that for any segment $[t, t')$ of T with $1 \le t < t' \le n + 1$,

(i) $\hat{w}_{t'}(\widehat{T}) > 0$, t and t' are switching points of $\widehat{T}$ (where $t' = n + 1$ is considered as a switching point), and $\widehat{T}$ contains at most $l = \left\lceil \frac{\log(t'-t)}{\lfloor \log(g+1) \rfloor} \right\rceil + 1$ segments in $[t, t')$;

(ii) if the switching points of $\widehat{T}$ in $[t, t')$ are $t_1 := t < t_2 < \cdots < t_{l'} < t_{l'+1} := t'$ then $l' \le l$ and for any nondecreasing function $f : [0, \infty) \to [0, \infty)$,

$$\sum_{i=1}^{l'} f(t_{i+1} - t_i) \le \sum_{i=0}^{l'-2} f\left(\frac{t' - t}{2^{i \lfloor \log(g+1) \rfloor}} \right) + f(t' - t) \tag{20}$$

$$\le \int_0^{\frac{\log(t'-t)}{\lfloor \log(g+1) \rfloor}} f\left(\frac{t' - t}{2^{x \lfloor \log(g+1) \rfloor}} \right) dx + 2f(t' - t) \tag{21}$$

where the second summation in (20) is empty if $l' = 1$.

Remark: Note that it is possible to obtain for l the less compact and harder-to-handle formula

$$l = \left\lceil \frac{\log\left(t' - t + \frac{1}{2^{\lfloor \log(g+1) \rfloor} - 1}\right) - \log\left(2^{\lfloor \log(g+1) \rfloor} - 1 + \frac{1}{2^{\lfloor \log(g+1) \rfloor} - 1}\right)}{\lfloor \log(g+1) \rfloor} \right\rceil + 1$$

by taking into account that the last segment $[t_l, t_{l+1})$ in the construction of the proof can always be defined to be of length at least $\lfloor \log(g+1) \rfloor 2^{u_l}$. Furthermore, for $g = 1$ it follows from [17] that the last term is not needed in (20), and hence it can be strengthened to

$$\sum_{c=1}^{l'} f(\hat{t}_{c+1} - \hat{t}_c) \le \sum_{c=0}^{\lfloor \log(t'-t) \rfloor} f(2^c). \tag{22}$$

Proof: Clearly, it is enough to define $\widehat{T}$ independently in each segment $[t, t')$ of T. We construct the switching points $t_1 < t_2 < \cdots < t_l$ of $\widehat{T}$ and an auxiliary variable $t_{l+1} \geq t'$ one by one such that $t_1 = t$, $t_l < t'$, and, defining u_j as the largest 2-power divisor of t_j,

$$u_{j+1} - u_j \geq \lfloor \log(g+1) \rfloor \tag{23}$$

for $j = 1, \ldots, l-1$. To simplify notation we also define $t_{l+1} = t'$. Let $t_1 = t$, and assume that we have already defined $t_1, \ldots, t_i$ satisfying (23) for $j = 1, \ldots, i-1$. Then the last segment is alive with positive probability at any time instant in $[t_i, t_i + g2^{u_i})$. If $t' \leq t_i + g2^{u_i}$, then let $t_{i+1} = t'$, else define u_{i+1} to be the largest nonnegative integer such that there is a $\tau \in [t_i + 1, t_i + g2^{u_i}]$ such that $2^{u_{i+1}}$ divides τ. Furthermore, let t_{i+1} denote the largest possible value of τ, that is,

$$t_{i+1} = \min\left\{t', \max\{\tau \in [t_i + 1, t_i + g2^{u_i}] : 2^{u_{i+1}} \text{ divides } \tau\}\right\}.$$

Then, whenever $t_{i+1} < t'$, $2^{u_{i+1}}$ is the largest 2-power divisor of t_{i+1}, and it is easy to see that $u_{j+1} \geq u_j + \lfloor \log(g+1) \rfloor$, proving (23) for $j = i$. Thus, from (23) we have

$$\begin{aligned}
t_{l+1} &\geq t + \sum_{i=1}^{l} 2^{u_i} = t + \sum_{i=1}^{l} 2^{u_1 + \sum_{j=2}^{i}(u_j - u_{j-1})} \\
&\geq t + \sum_{i=1}^{l} 2^{u_1 + \sum_{j=2}^{i} \lfloor \log(g+1) \rfloor} \geq t + \sum_{i=0}^{l-1} 2^{u_1 + i \lfloor \log(g+1) \rfloor} \\
&\geq t + 2^{u_1} \frac{2^{l \lfloor \log(g+1) \rfloor} - 1}{2^{\lfloor \log(g+1) \rfloor} - 1} \geq t + 2^{(l-1)\lfloor \log(g+1) \rfloor} \geq t'
\end{aligned}$$

where in the last step we used the definition of l. This finishes the proof of (i).

To prove (ii), we first show that the transition path $\widehat{T}$ constructed above satisfies (20). First notice that since $t + g2^{u_{l'-1}} \leq t_{l'-1} + g2^{u_{l'-1}} < t'$, we have $u_{l'-1} \leq \left\lfloor \log \frac{t'-t}{g} \right\rfloor$. Repeated application of (23) implies, for any $i = 1, \ldots, l'-1$,

$$u_i \leq \left\lfloor \log \frac{t'-t}{g} \right\rfloor - (l' - 1 - i) \lfloor \log(g+1) \rfloor$$

and

$$\begin{aligned}
t_{i+1} - t_i &\leq g 2^{\left\lfloor \log \frac{t'-t}{g} \right\rfloor - (l'-1-i)\lfloor \log(g+1) \rfloor} \\
&\leq g 2^{\log \frac{t'-t}{g} - (l'-1-i)\lfloor \log(g+1) \rfloor} = (t'-t) 2^{-(l'-1-i)\lfloor \log(g+1) \rfloor}.
\end{aligned}$$

Using the crude estimate $t' - t_l \leq t' - t$ finishes the proof of (20). The last inequality (21) holds trivially for $l = 1$, and holds for $l \geq 2$ since

$$\begin{aligned}
\sum_{i=0}^{l'-2} f\left(\frac{t'-t}{2^{i \lfloor \log(g+1) \rfloor}}\right) &= f(t'-t) + \sum_{i=1}^{l'-2} f\left(\frac{t'-t}{2^{i \lfloor \log(g+1) \rfloor}}\right) \\
&\leq f(t'-t) + \int_0^{\left\lceil \frac{\log(t'-t)}{\lfloor \log(g+1) \rfloor} \right\rceil - 1} f\left(\frac{t'-t}{2^{x \lfloor \log(g+1) \rfloor}}\right) dx \\
&\leq f(t'-t) + \int_0^{\frac{\log(t'-t)}{\lfloor \log(g+1) \rfloor}} f\left(\frac{t'-t}{2^{x \lfloor \log(g+1) \rfloor}}\right) dx.
\end{aligned}$$

■

Taking into account that $C(T_n) \leq C(\widehat{T}_n)$ if $\widehat{T}_n$ covers T_n, Lemma 3 trivially implies the following bounds.

Lemma 4: For any $T_n \in \mathcal{T}_n$ there exists a $\widehat{T}_n \in \mathcal{T}_n$ with $\hat{w}_n(\widehat{T}_n) > 0$ such that $\widehat{T}_n$ covers T_n and

$$C(T_n) \leq C(\widehat{T}_n) \leq (C(T_n)+1)L_{C(T_n),n} - 1 \tag{24}$$

where

$$L_{C,n} = \begin{cases} \left\lceil \frac{\log n}{\lfloor \log(g+1) \rfloor} \right\rceil + 1 & \text{if } C = 0, \\ \frac{\log \frac{n}{C+1}}{\lfloor \log(g+1) \rfloor} + 2 & \text{if } C \geq 1. \end{cases} \tag{25}$$

Proof: The lower bound is trivial, and the upper bound directly follows from Lemma 3 for $C(T_n) = 0$. For $C(T_n) \geq 1$ the upper bounds follow since on each segment of T_n we can define $\widehat{T}_n$ as in the proof of Lemma 3. Hence, if $T = (t_1, \ldots, t_C; n)$, then

$$\begin{aligned} C(\widehat{T}_n) + 1 &\leq \sum_{i=1}^{C+1} \left(\left\lceil \frac{\log(t_i - t_{i-1})}{\lfloor \log(g+1) \rfloor} \right\rceil + 1 \right) \leq \sum_{i=1}^{C+1} \left(\frac{\log(t_i - t_{i-1})}{\lfloor \log(g+1) \rfloor} + 2 \right) \\ &\leq (C+1) \left(\frac{\log \frac{n}{C+1}}{\lfloor \log(g+1) \rfloor} + 2 \right) \end{aligned}$$

where in the last step we used Jensen's inequality and the concavity of the logarithm. ■

We now apply the above construction and results to the weight function $\{w_t\} = \{w_t^{\mathcal{L}_1}\}$ to obtain our main theorem:

Theorem 2: Assume Algorithm 1 is run with weight function $\{\hat{w}_t^{\mathcal{L}_1}\}$ (derived from $\{w_t^{\mathcal{L}_1}\}$) with $g > 0$, based on a prediction algorithm that satisfies (5) for some $\rho_{\mathcal{E}}$. Let $L_{C,n}$ be defined by (25). If ℓ is convex in its first argument and takes values in the interval $[0,1]$ and $\eta_{t+1} \leq \eta_t$ for $t = 1, \ldots, n-1$, then for all n, the adaptive regret of the algorithm satisfies

$$R_n^a \leq L_{0,n} \rho_{\mathcal{E}} \left(\frac{n}{L_{0,n}} \right) + \sum_{t=1}^{n} \frac{\eta_t}{8} + \frac{r_n \left(L_{0,n} - 1 \right)}{\eta_n}$$

while for all n and any $T \in \mathcal{T}_n$ the tracking regret satisfies

$$\begin{aligned} \widehat{L}_n - L_n(T,a) \quad &\leq \quad L_{C(T),n}(C(T)+1) \rho_{\mathcal{E}} \left(\frac{n}{L_{C(T),n}(C(T)+1)} \right) \\ &\qquad + \sum_{t=1}^{n} \frac{\eta_t}{8} + \frac{r_n \left(L_{C(T),n}(C(T)+1) - 1 \right)}{\eta_n} \end{aligned} \tag{26}$$

where

$$r_n(C) = (C_n(T) + \epsilon) \ln n + \ln(1+\epsilon) - C_n(T) \ln \epsilon.$$

On the other hand, if ℓ is exp-concave for some $\eta > 0$ and we let $\eta_t = \eta$ for $t = 1, \ldots, n$ in Algorithm 1, then

$$R_n^a \leq L_{0,n} \rho_{\mathcal{E}} \left(\frac{n}{L_{0,n}} \right) + \frac{r_n \left(L_{0,n} - 1 \right)}{\eta_n}$$

while for any $T \in \mathcal{T}_n$ the tracking regret satisfies

$$\begin{aligned} \widehat{L}_n - L_n(T,a) &\le L_{C(T),n}(C(T)+1)\rho_{\mathcal{E}}\left(\frac{n}{L_{C(T),n}(C(T)+1)}\right) \\ &\quad + \frac{r_n\left(L_{C(T),n}(C(T)+1)-1\right)}{\eta_n}. \end{aligned} \tag{27}$$

Proof: First we show the bounds for the tracking regret. To prove the theorem, let $\widehat{T}_n$ be defined as in Lemma 1, and we bound the first and last terms on the right-hand side of (7) and (8) (with $\hat{w}_n^{\mathcal{L}_1}$ in place of w_n). Note that the conditions on $\rho_{\mathcal{E}}$ imply that $x\rho_{\mathcal{E}}(y/x)$ is a nondecreasing function of x for any fixed $y > 0$ (this follows since $\rho_{\mathcal{E}}(z)/z = (\rho_{\mathcal{E}}(z)-0)/(z-0)$ is a nonincreasing function of $z > 0$ by the concavity of $\rho_{\mathcal{E}}$, and hence $z\rho_{\mathcal{E}}(1/z)$ is nondecreasing). Combining this with the bounds on $C(T_n)$ in Lemma 4 implies

$$(C(\widehat{T}_n)+1)\rho_{\mathcal{E}}\left(\frac{n}{C(\widehat{T}_n)+1}\right) \le L_{C(T),n}(C(T)+1)\rho_{\mathcal{E}}\left(\frac{n}{L_{C(T),n}(C(T)+1)}\right).$$

The last term $(1/\eta_n)\ln(1/\hat{w}_n^{\mathcal{L}_1}(\widehat{T}_n)$ in (7) and (8) can be bounded by noting that $1/\hat{w}_n^{\mathcal{L}_1}(\widehat{T}_n) \le 1/w_n^{\mathcal{L}_1}(\widehat{T}_n)$ by (19) and the latter can be bounded using (14); this is given by r_n. This finishes the proof of the tracking regret bounds.

Next we prove the bounds for the adaptive regret. Assume we want to bound the regret of our algorithm in a segment $[t,t')$ with $1 \le t < t' \le n+1$. By Lemma 3 there exists a transition path $\widehat{T}_n$ such that it has a switching point at t, has at most $l = \left\lceil \frac{\log(t'-t)}{\lfloor \log(g+1) \rfloor} \right\rceil + 1 \le L_{0,n}$ segments in $[t,t')$, and $\hat{w}_n(\widehat{T}_n) > 0$. Furthermore, let $(\tilde{T}_n, a)$ be a meta expert that follows the optimal base expert $\tilde{\imath}$ in the time interval $[t,t')$ and outside this interval it predicts the same as algorithm $\mathcal{A}$ applied on $\widehat{T}_n$. Now let $\hat{t}_1, \hat{t}_2, \ldots, \hat{t}_{\widehat{C}}$ denote the switching points of $\widehat{T}_n$ in $[t+1,t')$ where $\widehat{C} < l$, and let $\hat{t}_0 = t$ and $\hat{t}_{\widehat{C}+1} = t'$.

Repeating the proof of Lemma 1 to bound the difference $\widehat{L}_n - L_n(\tilde{T}_n, a)$, we obtain modified versions of (7) and (8), where the first terms are changed. Indeed, instead of (9) we have

$$\begin{aligned} L_n(\mathcal{A}, \widehat{T}_n) - L_n(\tilde{T}_n, a) &= \sum_{c=0}^{\widehat{C}}\Big(L_{\mathcal{A}}(\hat{t}_c, \hat{t}_{c+1}) - L_{\tilde{\imath}}(\hat{t}_c, \hat{t}_{c+1})\Big) \\ &\le \sum_{c=0}^{\widehat{C}} \rho_{\mathcal{E}}(\hat{t}_{c+1} - \hat{t}_c) \le l\rho_{\mathcal{E}}\left(\frac{t'-t}{l}\right) \le L_{0,n}\rho_{\mathcal{E}}\left(\frac{n}{L_{0,n}}\right). \end{aligned} \tag{28}$$

This proves the adaptive regret bounds. ■

Remark: Note that the tracking regret can be trivially bounded by $(C(T)+1)$ times the adaptive regret (as suggested by [19]) but the tracking regret bounds are clearly better than this. The difference is more pronounced for the case of the convex and bounded loss function, where the constant of the main term is affected, while only lower order terms are affected in case of an exp-concave loss function. In either case, the bounds of Theorem 2 slightly improve those of [19] for the adaptive regret.

E. Exponential weighting

We now apply Theorem 2 to the case where $\mathcal{A}$ is the exponentially weighted average forecaster and the set of base experts is of size N, and discuss the obtained bounds (for simplicity we assume $C(T) \geq 1$, but $C(T) = 0$ would just slightly change the presented bounds). In this case, if ℓ is convex and bounded, then by (3) the regret of $\mathcal{A}$ is bounded by $\rho_{\mathcal{E}}(n) = \sqrt{n \ln N}$. Setting $\eta_t \equiv \phi \ln n / \sqrt{n}$ for some $\phi > 0$ (η_t is independent of $C(T)$ but depends on the time horizon n), the bound (26) becomes, for $g = O(1)$,

$$\begin{aligned} \widehat{L}_n - L_n(T,a) \ &\leq\ \sqrt{n(C(T)+1)\left(\frac{\log n}{\lfloor \log(g+1) \rfloor} + 2\right)\ln N} \\ &\quad + \frac{\phi\sqrt{n}\ln n}{8} + \frac{(C+1)\sqrt{n}}{\phi}\left(\frac{\log n}{\lfloor \log(g+1) \rfloor} + 2\right) + O\left(\frac{\sqrt{n}}{\ln n}\right) \ . \end{aligned}$$

Furthermore, if an upper bound C on the complexity (number of switches) of the meta experts in the reference class is known in advance, then η_t can be set as a function of $C \geq C(T)$ as well, resulting $\eta_t \equiv \sqrt{8(C+1)\ln n\left(\frac{\log n}{\lfloor \log(g+1) \rfloor} + 2\right)/n}$, in which case the bound (26) becomes

$$\begin{aligned} \widehat{L}_n - L_n(T,a) \ &\leq\ \sqrt{n(C(T)+1)\left(\frac{\log n}{\lfloor \log(g+1) \rfloor} + 2\right)\ln N} \\ &\quad + \sqrt{\frac{n(C+1)\left(\frac{\log n}{\lfloor \log(g+1) \rfloor} + 2\right)\ln n}{2}} + O\left(\sqrt{\frac{n}{(C+1)\ln n\left(\frac{\log n}{\lfloor \log(g+1) \rfloor} + 2\right)}}\right) \ . \end{aligned}$$

We note that these bounds are only larger by a factor of $O(\sqrt{\ln n})$ than the ones resulting from earlier algorithms [11], [12], [27] which have complexity $O(n^2)$. In some applications, such as online quantization [27], the number of base experts N depends on the time horizon n in a polynomial fashion, that is, $N \sim n^{\beta}$ for some $\beta > 0$. In such cases the upper bound becomes $O((C(T)+1)\sqrt{n \ln^2 n})$ if the number of switches is unknown, and $O(\sqrt{(C(T)+1) n \ln^2 n})$ if the maximum number of switches $C(T)$ is known in advance. This bound is within a factor of $O(\sqrt{\log n})$ of the best achievable regret for this case.

Next we observe that at the price of a slight increase of computational complexity, regret bounds of the optimal order can be obtained. Indeed, setting $g = 2n^{\gamma} - 1$ for some $\gamma \in (0,1)$ and $\eta_t \equiv \phi\sqrt{\frac{(2+1/\gamma)\ln n}{n}}, \phi > 0$ independently of the maximum number of switches,

$$\begin{aligned} &\widehat{L}_n - L_n(T,a) \\ &\quad \leq\ \sqrt{n(C(T)+1)\ln N\left(\frac{1}{\gamma} + 2\right)} + \left(\frac{\phi}{8} + \frac{C+1}{\phi}\right)\sqrt{\left(\frac{1}{\gamma} + 2\right) n \ln n} + O\left(\sqrt{\frac{n}{\ln n}}\right). \end{aligned}$$

If η_t is optimized for an a priori known bound $C \geq C(T)$, then we get

$$\begin{aligned} &\widehat{L}_n - L_n(T,a) \\ &\qquad \leq\ \sqrt{n(C(T)+1)\left(\frac{1}{\gamma} + 2\right)}\left(\sqrt{\ln N} + \sqrt{\frac{\ln n}{2}}\right) + O\left(\sqrt{\frac{n}{(C+1)\ln n}}\right). \end{aligned}$$

These bounds are of the same order as the ones achievable with the quadratic complexity algorithms [11], [24], [27], but the complexity of our algorithm is only $O(n^\gamma \log n)$ times larger than that of running $\mathcal{A}$ (which is typically linear in n). Thus, in a sense the complexity of our algorithm can get very close to linear while guaranteeing a regret of optimal order. (Note however, that a factor $1/\sqrt{\gamma}$ appears in the regret bounds so setting γ very small comes at a price.)

A similar behavior is observed for exp-concave loss functions. Indeed, if ℓ is exp-concave and $\mathcal{A}$ is the exponentially weighted average forecaster, then by (4) the regret of $\mathcal{A}$ is bounded by $\rho_{\mathcal{E}}(n) = \frac{\log N}{\eta}$. In this case, for $g = O(1)$, the bound (27) becomes

$$\widehat{L}_n - L_n(T,a) \quad \le \quad \frac{(C(T)+1)\left(\frac{\log \frac{n}{C(T)+1}}{\lfloor \log(g+1) \rfloor} + 2\right)}{\eta}\,(\ln N + \ln n) + O(1).$$

which is a factor of $O(\ln n)$ larger than the existing bounds [10]–[12], [14], [24] valid for algorithms having complexity $O(n^2)$. Note that in this case the algorithm is strongly sequential as its parametrization is independent of the time horizon n. For $g = 2n^\gamma - 1$, we obtain a bound of optimal order:

$$\widehat{L}_n - L_n(T,a) \quad \le \quad \frac{(C(T)+1)\left(\frac{1}{\gamma} + 2\right)}{\eta}\,(\ln N + \ln n) + O(1).$$

F. The weight function w^{KT}

In this section we analyze the performance of Algorithm 1 for the case when the "Krichevsky-Trofimov" weight function w^{KT} is used. Our analysis is based on part (ii) of Lemma 3, following ideas of Willems and Krom [17] who only considered the logarithmic loss. Applying the weight function $\hat{w}^{KT}$ (derived from w^{KT}), this analysis improves the constants relative to Theorem 2 for small values of g, although the resulting bound has a less compact form. Nevertheless, in some special situations the bounds can be expressed in a simple form. This is the case for the logarithmic loss, where, for the special choice $g = 1$, applying (22), the new bound now achieves that of [17] proved for the same algorithm. The idea is that in the proof of Theorem 2 the concavity of $\rho_{\mathcal{E}}$ was used to get simple bonds on sums which are sharp if the segments are of (approximately) equal length. However, in our construction the length of the sub-segments (corresponding to the same segment of the original transition path), or more precisely, their lower bounds, grow exponentially according to (23). This makes it possible to improve the upper bounds in Theorem 2. It is interesting to note that the weight functions $w^{\mathcal{L}_1}$ and $w^{\mathcal{L}_2}$ give better bounds for $g = n^\gamma$, where the segment lengths are approximately equal, while the large differences in the segment lengths for $g = O(1)$ can be exploited by the weight function w^{KT}.

To obtain "almost closed-form" regret bounds for a general $\rho_{\mathcal{E}}$, we need the following technical lemma.

Lemma 5: Assume $f : [1,\infty) \to (0,\infty)$ is a differentiable function and $G \ge 1$. Define $F : [1,\infty) \to [0,\infty)$ by

$$F(s) = \int_0^{\frac{\log s}{G}} f\left(\frac{s}{2^{cG}}\right)\, dc$$

for all $s \geq 1$. Then the second derivative of F is given by

$$F''(s) = \frac{f'(s)}{sG\ln 2} - \frac{f(s)}{s^2 G \ln 2}.$$

Therefore, F is concave on $[1, \infty)$ if $sf'(s) \leq f(s)$ for all $s \geq 1$.

Proof: First note that, since $2^{cG} = s$ for $c = \frac{\log s}{G}$, Leibniz's integral rule gives

$$F'(s) = \frac{f(1)}{sG\ln 2} + \int_0^{\frac{\log s}{G}} f'\left(\frac{s}{2^{cG}}\right) 2^{-cG} \, dc = \frac{f(1) - f(1) + f(s)}{sG\ln 2} = \frac{f(s)}{sG\ln 2}$$

since

$$-\frac{\partial}{\partial c}\frac{f\left(s2^{-cG}\right)}{sG\ln 2} = f'\left(s2^{-cG}\right)2^{-cG}.$$

Differentiating F' gives the desired result. ■

Next we give an improvement of Theorem 2 for small values of g.

Theorem 3: Assume $\rho_{\mathcal{E}}(x)$ is differentiable and satisfies $\rho_{\mathcal{E}}(x) \geq x\rho'_{\mathcal{E}}(x)$ for all $x \geq 1$, Algorithm 1 is run with weight function $\{\hat{w}_t^{KT}\}$. Let

$$S(C,n) \;=\; (C+1)\int_0^{\frac{\log\frac{n}{C+1}}{\lfloor\log(g+1)\rfloor}} \rho_{\mathcal{E}}\left(\frac{n}{C+1}2^{-c\lfloor\log(g+1)\rfloor}\right) dc + 2(C+1)\rho_{\mathcal{E}}\left(\frac{n}{C+1}\right)$$

and

$$\begin{aligned}\bar{r}_n(C) \;=\; \frac{(C+1)\ln 2}{4}\Bigg(&\frac{\log^2\frac{n}{C+1}}{\lfloor\log(g+1)\rfloor} + \left(4 + \frac{4}{\lfloor\log(g+1)\rfloor}\right)\log\frac{n}{C+1} \\ &+ \lfloor\log(g+1)\rfloor + 8\Bigg).\end{aligned}$$

If ℓ is convex in its first argument and takes values in the interval $[0,1]$, and $\eta_{t+1} \leq \eta_t$ for $t = 1, \ldots, n-1$, then for all n the adaptive regret of the algorithm satisfies

$$R_n^a \leq S(0,n) + \sum_{t=1}^{n}\frac{\eta_t}{8} + \frac{\bar{r}_n(0)}{\eta_n}$$

while for any $T \in \mathcal{T}_n$ the tracking regret satisfies, for all n,

$$\widehat{L}_n - L_n(T,a) \leq S(C,n) + \sum_{t=1}^{n}\frac{\eta_t}{8} + \frac{\bar{r}_n(C)}{\eta_n}. \tag{29}$$

On the other hand, if ℓ is exp-concave for the value of η and $\eta_t = \eta$ for $t = 1, \ldots, n$ in Algorithm 1, then

$$R_n^a \leq S(0,n) + \frac{\bar{r}_n(0)}{\eta_n}$$

while for any $T \in \mathcal{T}_n$ the tracking regret satisfies

$$\widehat{L}_n - L_n(T,a) \leq S(C,n) + \frac{\bar{r}_n(C)}{\eta_n}. \tag{30}$$

Proof: We proceed similarly to the proof of Theorem 2 by first applying Lemma 1. However, the resulting two terms are now bounded using Lemma 3 (ii) instead of Jensen's inequality, which allows us to make use of the potentially large differences in the segment lengths.

For any transition path $T = (t_1, \dots, t_C) \in \mathcal{T}_n$ let $\widehat{T} = (\hat{t}_1, \dots, \hat{t}_{\widehat{C}}) \in \mathcal{T}_n$ denote the transition path defined by Lemma 3 with $\hat{w}_n^{KT}(\widehat{T}) > 0$. The first term of the first upper bound given in Lemma 1 can be bounded as follows: for any segment $[t_c, t_{c+1}) = [\hat{t}_{\hat{c}}, \hat{t}_{\hat{c}'})$ of T, Lemma 3 (i) and (21) yield

$$\sum_{i=\hat{c}}^{\hat{c}'-1} \rho_{\mathcal{E}}(\hat{t}_{i+1} - \hat{t}_i) \le \int_0^{\frac{\log(t_{c+1}-t_c)}{\lfloor \log(g+1) \rfloor}} \rho_{\mathcal{E}}\left(\frac{t_{c+1}-t_c}{2^{c\lfloor \log(g+1)\rfloor}}\right)\, dc + 2\rho_{\mathcal{E}}(t_{c+1}-t_c).$$

Since the right-hand side of the above equation is a concave function of $s = t_{c+1} - t_c$ by Lemma 5 and the conditions on $\rho_{\mathcal{E}}$, Jensen's inequality implies

$$\begin{aligned}
&\sum_{i=0}^{\widehat{C}} \rho_{\mathcal{E}}(\hat{t}_{i+1} - \hat{t}_i) \\
&\quad = \sum_{c=0}^{C} \sum_{i=\hat{c}}^{\hat{c}'-1} \rho_{\mathcal{E}}(\hat{t}_{i+1} - \hat{t}_i) \\
&\quad \le \sum_{c=0}^{C} \left(\int_0^{\frac{\log(t_{c+1}-t_c)}{\lfloor \log(g+1) \rfloor}} \rho_{\mathcal{E}}\left(\frac{t_{c+1}-t_c}{2^{c\lfloor \log(g+1)\rfloor}}\right)\, dc + 2\rho_{\mathcal{E}}(t_{c+1}-t_c) \right) \\
&\quad \le (C+1) \int_0^{\frac{\log \frac{n}{C+1}}{\lfloor \log(g+1) \rfloor}} \rho_{\mathcal{E}}\left(\frac{n}{C+1} \cdot 2^{-c\lfloor \log(g+1)\rfloor}\right)\, dc + 2(C+1)\rho_{\mathcal{E}}\left(\frac{n}{C+1}\right). \qquad (31)
\end{aligned}$$

The weight function can be bounded in a similar way. By the standard bound (13) on the Krichevsky-Trofimov estimate, we have

$$\ln \frac{1}{\hat{w}_n^{KT}(\widehat{T})} \le \ln \frac{1}{w_n^{KT}(\widehat{T})} \le \sum_{c=0}^{\widehat{C}} \left(\frac{1}{2} \ln(\hat{t}_{c+1} - \hat{t}_c) + \ln 2 \right). \qquad (32)$$

Applying (20) for a segment $[t_c, t_{c+1}) = [\hat{t}_{\hat{c}}, \hat{t}_{\hat{c}'})$ of T yields

$$\begin{aligned}
&\sum_{i=\hat{c}}^{\hat{c}'-1} \left(\frac{1}{2} \ln(\hat{t}_{i+1} - \hat{t}_i) + \ln 2 \right) \\
&\quad \le \sum_{i=0}^{\left\lceil \frac{\log(t_{c+1}-t_c)}{\lfloor \log(g+1) \rfloor} \right\rceil - 1} \left(\frac{1}{2} \ln\left(\frac{t_{c+1}-t_c}{2^{i\lfloor \log(g+1)\rfloor}} \right) + \ln 2 \right) + \frac{1}{2}\ln(t_{c+1}-t_c) + \ln 2 \\
&\quad = \frac{\ln 2}{2} \left\lceil \frac{\log(t_{c+1}-t_c)}{\lfloor \log(g+1) \rfloor} \right\rceil \left(\log(t_{c+1}-t_c) - \frac{\left\lceil \frac{\log(t_{c+1}-t_c)}{\lfloor \log(g+1) \rfloor} \right\rceil - 1}{2} \lfloor \log(g+1) \rfloor + 2 \right) \\
&\qquad + \frac{1}{2}\ln(t_{c+1}-t_c) + \ln 2 \\
&\quad \le \frac{\ln 2}{4} \left(\frac{\log^2(t_{c+1}-t_c)}{\lfloor \log(g+1) \rfloor} + \left(4 + \frac{4}{\lfloor \log(g+1) \rfloor} \right) \log(t_{c+1}-t_c) + \lfloor \log(g+1) \rfloor + 8 \right)
\end{aligned}$$

where in the last step we bounded the ceiling function from above and from below, as appropriate. Furthermore, it is easy to check that the last expression above is concave in $s = t_{c+1} - t_c$. Therefore, combining it with (32), applying Jensen's inequality, we obtain

$$\ln \frac{1}{\hat{w}_n^{KT}(\widehat{T})} \leq \bar{r}_n(C).$$

Applying this bound and (31) in Lemma 1 yields the statements of the theorem. ■

We now apply Theorem 3 to the exponentially weighted average predictor. For convex loss functions we have $\rho_{\mathcal{E}}(n) = \sqrt{n \ln N}$. Assuming $g = O(1)$, if $\eta_t \equiv \phi \sqrt{\frac{2\ln 2}{n\lfloor \log(g+1)\rfloor}} \log n, \phi > 0$ (i.e., η_t is independent of the number of switches $C(T)$), we obtain

$$\begin{aligned} \widehat{L}_n - L_n(T,a) &\leq 2\sqrt{(C(T)+1)n\ln N}\left(1 + \frac{1 - \sqrt{\frac{C+1}{n}}}{\lfloor \log(g+1)\rfloor \ln 2}\right) \\ &\quad + \frac{\phi + \frac{C+1}{\phi}}{4} \log n \sqrt{\frac{n \ln 2}{2\lfloor \log(g+1)\rfloor}} + o\big((C+1)\sqrt{n}\big). \end{aligned}$$

Optimizing η_t as a function of an upper bound C on the number of switches yields

$$\begin{aligned} \widehat{L}_n - L_n(T,a) &\leq 2\sqrt{(C(T)+1)n\ln N}\left(1 + \frac{1 - \sqrt{\frac{C+1}{n}}}{\lfloor \log(g+1)\rfloor \ln 2}\right) \\ &\quad + \sqrt{\frac{(C+1)n \log^2 \frac{n}{C+1} \ln 2}{8\lfloor \log(g+1)\rfloor}} + o\big(\sqrt{(C+1)n}\big). \end{aligned}$$

Note that if $N = O(n^\beta)$ for some $\beta > 0$, the first term is asymptotically negligible compared to the second in the above bounds. For example, if η is set independently of C, we obtain

$$\widehat{L}_n - L_n(T,a) \leq \frac{\phi + \frac{C+1}{\phi}}{4} \log n \sqrt{\frac{n \ln 2}{2\lfloor \log(g+1)\rfloor}} + o\big((C+1)\sqrt{n}\big).$$

On the other hand, if $g = 2n^\gamma - 1$, the bound becomes

$$\begin{aligned} \widehat{L}_n - L_n(T,a) &\leq 2\sqrt{(C(T)+1)n\ln N}\left(1 + \frac{1 - \sqrt{\frac{C+1}{n}}}{\gamma \ln n}\right) \\ &\quad + \frac{\phi + \frac{C+1}{\phi}}{8} \sqrt{2n \ln n \left(4 + \gamma + \frac{1}{\gamma}\right)} + O\left(\sqrt{\frac{n}{\ln n}}\right) \end{aligned}$$

when η is set independently of C.

For exp-concave loss functions we have, for $g = O(1)$,

$$\widehat{L}_n - L_n(T,a) \leq \frac{C+1}{4\eta}\left(\frac{\log \frac{n}{C+1}}{\lfloor \log(g+1)\rfloor} + 2\right)\left(4\ln N + \ln \frac{n}{C+1}\right) + O(C \ln n)$$

while if $g = 2n^\gamma - 1$ we get

$$\widehat{L}_n - L_n(T,a) \le \frac{C+1}{4\eta}\left(4\left(\frac{1}{\gamma}+2\right)\ln N + \left(4+\gamma+\frac{1}{\gamma}\right)\ln n\right) + O(C).$$

Note that for both types of loss functions we have a clear improvement relative to Theorem 2, where we used the weight function $w^{\mathcal{L}_1}$, for the case when $g = O(1)$. However, no such distinction can be made for $g = 2n^\gamma - 1$. Indeed, for convex loss functions constant multiplicative changes in η vary the exact form of the factor $(C+a)/b$, with constants $a, b > 0$ in the second term, and, consequently, the order of the bounds depends on the relative size of C, while, for example, the value of η determines the order of the bounds for exp-concave losses, e.g., constructing the weigh function $\hat{w}$ from $w^{\mathcal{L}_1}$ is better for $\gamma \ge 1/3$. Also note that the above bounds for $g = 3$ and $g = 4$ have improved leading constant compared to [18] and [28], respectively.

III. Randomized prediction

The results of the previous section may be adapted to the closely related model of randomized prediction. In this framework, the decision maker plays a repeated game against an adversary as follows: at each time instance $t = 1, \ldots, n$, the decision maker chooses an action I_t from a finite set, say $\{1, \ldots, N\}$ and, independently, the adversary assigns losses $\ell_{i,t} \in [0,1]$ to each action $i = 1, \ldots, n$. The goal of the decision maker is to minimize the cumulative loss $\widehat{L}_n = \sum_{t=1}^n \ell_{I_t,t}$.

Similarly to the previous section, the decision maker may try to compete with the best sequence of actions that can change actions a limited number of time instants. More precisely, the set of base experts is $\mathcal{E} = \{1, \ldots, N\}$ and as before, we may define a meta expert that changes base experts C times by a transition path $T = (t_1, \ldots, t_C; n)$ and a vector of actions $a = (i_0, \ldots, i_C)$, where $t_0 := 1 < t_1 < \ldots < t_C < t_{C+1} := n+1$ and $i_j \in \{1, \ldots, N\}$. The total loss of the meta expert indexed by (T, a), accumulated during n rounds, is

$$L_n(T,a) = \sum_{c=0}^{C} L_{i_c}(t_c, t_{c+1}) \quad \text{with} \quad L_{i_c}(t_c, t_{c+1}) = \sum_{t=t_c}^{t_{c+1}-1} \ell_{i_c,t} \ .$$

There are two differences relative to the setup considered earlier. First, we do not assume that the loss function satisfies special properties such as convexity in the first argument (although we do require that it be bounded). Second, we do not assume in the current setup that the action space is convex, and so a convex combination of the experts' advice is not possible. On the other hand, similar results as before can be achieved if the decision maker may randomize its decisions, and in this section we deal with this situation.

In randomized prediction, before taking an action, the decision maker chooses a probability distribution $\boldsymbol{p}_t$ over $\{1, \ldots, N\}$ (a vector in the probability simplex Δ_N in $\mathbb{R}^N$), and chooses an action I_t distributed according to $\boldsymbol{p}_t$ (conditionally, given the past actions of the decision maker and the losses assigned by the adversary).

Note that now both $\widehat{L}_n$ and $L_n(T,a)$ are random variables not only because the decision takes randomized decisions but also because the losses set by the adversary may depend on past randomized choices of the decision maker. (This model is known as the "non-oblivious adversary".) We may define the *expected loss* of the decision maker by

$$\overline{\ell}_t(\boldsymbol{p}_t) = \sum_{i=1}^{N} p_{i,t}\ell_{i,t}$$

where $p_{i,t}$ denotes the i-th component of $\boldsymbol{p}_t$.

For details and discussion of this standard model we refer to [1, Section 4.1]. In particular, since the results presented in Section I can be extended to time-varying loss functions and since $\overline{\ell}_t$ is a linear (convex) function, it can be shown that regret bounds of any forecaster in the model of Section I can be extended to the sequence of loss functions $\overline{\ell}_t$. That is, the bounds can be converted into bounds for the expected regret of a randomized forecaster. Furthermore, it is shown in [1, Lemma 4.1] how such bounds in expectation can be converted to bound that hold with high probability.

For example, a straightforward combination of [1, Lemma 4.1] and Theorem 2 implies the following. Consider a prediction algorithm $\mathcal{A}$ defined in the model of Section II-A, that chooses an action in the decision space $\mathcal{D} = \Delta_N$ and suppose that it satisfies a regret bound of the form (5) under the loss function $\overline{\ell}_t(\boldsymbol{p}_t)$. Algorithm 2 below, which is a variant of Algorithm 1, converts $\mathcal{A}$ into a forecaster under the randomized model. At each time instant t, the algorithm chooses, in a randomized way, a transition path $T = (t_1, \ldots, t_C; t) \in \mathcal{T}_t$, and uses the distribution $\boldsymbol{p}_{\mathcal{A},t}(\tau_t(T))$ that $\mathcal{A}$ would choose, had it been started at time $\tau_t(T)$, the time of the last change in the path T up to time t. In the definition of the algorithm

$$\overline{L}_t(\mathcal{A}, T) = \sum_{c=0}^{C} \overline{L}_{\mathcal{A}}(t_c, t_{c+1})$$

denotes the cumulative expected loss of algorithm $\mathcal{A}$, where we define $t_0 = 1$ and $t_{c+1} = t + 1$, and

$$\overline{L}_{\mathcal{A}}(t_c, t_{c+1}) = \sum_{\tau=t_c}^{t_{c+1}-1} \overline{\ell}_\tau(\boldsymbol{p}_{\mathcal{A},\tau}(t_c))$$

is the cumulative expected loss suffered by $\mathcal{A}$ in the time interval $[t_c, t_{c+1})$ with respect to $\overline{\ell}_\tau$ for $\tau \in [t_c, t_{c+1})$.

Algorithm 2 Randomized tracking algorithm.

Input: Prediction algorithm $\mathcal{A}$, weight function $\{w_t; t = 1, \ldots, n\}$, learning parameters $\eta_t > 0, t = 1, \ldots, n$.

For $t = 1, \ldots, n$ choose $T \in \mathcal{T}_t$ according to the distribution

$$q_t(T) = \frac{w_t(T) e^{-\eta_t \overline{L}_{t-1}(\mathcal{A}, T_{t-1})}}{\sum_{T' \in \mathcal{T}_t} w_t(T') e^{-\eta_t \overline{L}_{t-1}(\mathcal{A}, T'_{t-1})}} ,$$

choose $\boldsymbol{p}_t = \boldsymbol{p}_{\mathcal{A},t}(\tau_t(T))$, and pick $I_t \sim \boldsymbol{p}_t$.

Corollary 1: Suppose $\ell_{i,t} \in [0, 1]$ for all $i = 1, \ldots, N$ and $t = 1, \ldots, n$, and $\mathcal{A}$ satisfies (5) with respect to the loss function $\{\ell_t\}$. Assume Algorithm 2 is run with weight function $\{\hat{w}^{\mathcal{L}_1}\}$ for some

$\epsilon > 0$. Let $\delta \in (0,1)$. For any $T \in \mathcal{T}_n$, the regret of the algorithm satisfies, with probability at least $1-\delta$,

$$\begin{aligned}\widehat{L}_n - L_n(T,a) \quad &\le \quad L_{C(T),n}(C(T)+1)\rho_{\mathcal{E}}\left(\frac{n}{L_{C(T),n}(C(T)+1)}\right) + \sum_{t=1}^{n}\frac{\eta_t}{8} \\ &\quad + \frac{r_n\left(L_{C(T),n}(C(T)+1)-1\right)}{\eta_n} + \sqrt{\frac{n}{2}\ln\frac{1}{\delta}}\ .\end{aligned}$$

where $r_n(C)$ and $L_{C,n}$ are defined as in Theorem 2.

Proof: First note that Theorem 2 can easily be extended to time-varying loss functions (in fact, Lemma 1, and consequently Theorem 2, uses the bound (2) which allows time-varying loss functions). Combining the obtained bound for the expected loss with [1, Lemma 4.1] proves the corollary. ■

IV. Examples

In this section we apply the results of the paper for a few specific examples.

Example 1 (Krichevsky-Trofimov mixtures): Assume $\mathcal{D} = \mathcal{E} = (0,1)$ and $\mathcal{Y} = \{0,1\}$, and consider the logarithmic loss defined as $\ell(p,y) = -\mathbb{I}_{y=1}\log p - \mathbb{I}_{y=0}\log(1-p)$. As mentioned before, the logarithmic loss is exp-concave with $\eta \le 1$, and hence we choose $\eta = 1$. This loss plays a central role in data compression. In particular, if a prediction method achieves, on a particular binary sequence $y^n = (y_1,\ldots,y_n)$, a loss $\widehat{L}_n$, then using arithmetic coding the sequence can be described with at most $L_n + 2$ bits [29]. We note that the choice of the expert class $\mathcal{E} = (0,1)$ corresponds to the situation where the sequence y^n is encoded using an i.i.d. coding distribution. Competing against the expert class $\mathcal{E} = (0,1)$ also has a probabilistic interpretation: it is equivalent to minimizing the worst case maximum coding redundancy relative to the class of i.i.d. source distributions on $\{0,1\}^n$.

Let $n_0(t) = \sum_{\tau=1}^{t}\mathbb{I}_{y_\tau=0}$ and $n_1(t) = \sum_{\tau=1}^{t}\mathbb{I}_{y_\tau=1}$ denote the number of 0s and 1s in y^t, respectively. Then the loss of an expert $\theta \in (0,1)$ at time t is

$$L_{\theta,t} = -\log\left((1-\theta)^{n_0(t)}\theta^{n_1(t)}\right) = -n_0(t)\log(1-\theta) - n_1(t)\log\theta$$

which is the negative log-probability assigned to y^t by a memoryless binary Bernoulli source generating 1s with probability θ. The Krichevsky-Trofimov forecaster is an exponentially weighted average forecaster over all experts $\theta \in \mathcal{E}$ using initial weights $1/(\pi\sqrt{\theta(1-\theta)})$ (i.e., the Beta$(1/2,1/2)$ distribution) defined as

$$p_t^{KT}(y^{t-1}) = \int_0^1 \frac{e^{-L_{\theta,t-1}}}{\pi\sqrt{\theta(1-\theta)}}\,d\theta = \int_0^1 \frac{(1-\theta)^{n_0(t-1)}\theta^{n_1(t-1)}}{\pi\sqrt{\theta(1-\theta)}}\,d\theta.$$

It is well known that p_t^{KT} can be computed efficiently as $p_t^{KT}(y^{t-1}) = (n_1(t-1)+1/2)/t$. The performance of the Krichevsky-Trofimov mixture forecaster can be bounded as

$$R_n \le \frac{1}{2}\ln n + \ln 2.$$

In this framework, a meta expert based on the base expert class $\mathcal{E}$ is allowed to change $\theta \in \mathcal{E}$ a certain number of times. In the probabilistic interpretation, this corresponds to the problem of coding a piecewise i.i.d. source [10], [11], [15]–[17]. If we apply Algorithm 1 to this problem with $\hat{w}^{KT}$, we can improve upon Theorem 3 by using $\bar{r}_n(C)$ instead of $S(C,n)$ in the bound (note that $\bar{r}_n(C)$ was obtained by calculating the Krichevsky-Trofimov bound for the transition probabilities), and obtain, for any transition path $T \in \mathcal{T}_n$ and meta expert (T,a)

$$\widehat{L}_n - L_n(T,a) \quad \le \quad 2\bar{r}_n(C(T)) = \frac{(C(T)+1)\ln 2}{2} \frac{\log^2 \frac{n}{C(T)+1}}{\lfloor \log(g+1) \rfloor} + O((C(T)+1)\log n).$$

For $g=1$, this bound recovers that of [17] (at least in the leading term), and improves the leading constant for $g=3$ and $g=4$ when compared to [18] and [19], respectively.

On the other hand, for $g = 2n^{\gamma} - 1$, $\gamma > 0$, using with $\hat{w}^{\mathcal{L}_1}$ in Algorithm 1, Theorem 3 implies

$$\widehat{L}_n - L_n(T,a) \le \frac{3(C+1)}{2}\left(\frac{1}{\gamma}+2\right)\ln n + O(1).$$

This bound achieves the optimal $O(\ln n)$ order for any $\gamma > 0$; however, with increased leading constant. On the negative side, for specific choices of γ our algorithm does not recover the best leading constants now in the literature (partly due to the common bounding technique for all γ): If $\gamma = 1/2$, our bound is a constant factor worse than those of [15] and [16] which have the same $O(n^{3/2})$ complexity (disregarding logarithmic factors); on the other hand, in case $\gamma = 1$ our algorithm is identical to the $O(n^2)$ complexity algorithm of Shamir and Merhav [11], and hence an optimal bound can be proved for $\hat{w}^{\mathcal{L}_1}$ (and for $\hat{w}^{\mathcal{L}_2}$), as done in [11] achieving Merhav's lower bound [30].

Example 2 (Tracking structured classes of base experts): In recent years a significant body of research has been devoted to prediction problems in which the forecaster competes with a large but structured class of experts. We refer to [1], [5], [7], [8], [20], [27], [31], [32] for an incomplete but representative list of papers. A quite general framework that has been investigated is the following: a base expert is represented by a d-dimensional binary vector $v \in \{0,1\}^d$. Let $\mathcal{E} \subset \{0,1\}^d$ be the class of experts. The decision space $\mathcal{D}$ is the convex hull of $\mathcal{E}$, so the forecaster chooses, at each time instance $t = 1,\dots,n$, a convex combination $\widehat{p}_t = \sum_{v\in\mathcal{E}} \pi_{v,t} v \in \mathcal{D} \subset [0,1]^d$. The outcome space is $\mathcal{Y} = [0,1]^d$ and if the outcome is $y_t \in \mathcal{Y}$, then the loss of expert v is $\ell(v,y_t) = v^T y_t$, the standard inner product of v and y_t. The loss of the forecaster equals $\ell(\widehat{p}_t, y_t) = \sum_{v\in\mathcal{E}} \pi_{v,t} v^T y_t$. [8] introduces a general prediction algorithm, called "Component Hedge," that achieves a regret

$$\sum_{t=1}^{n} \ell(\widehat{p}_t, y_y) - \min_{v\in\mathcal{E}} \sum_{t=1}^{n} \ell(v,y_t) \le d\sqrt{2Kn\ln(d/K)} + dK\ln(d/K)$$

where $K = \max_{v\in\mathcal{E}} \|v\|_1$. What makes Component Hedge interesting, apart from its good regret guarantee, is that for many interesting classes of base experts it can be calculated in time that is polynomial in d, even when $\mathcal{E}$ has exponentially many experts. We refer to [8] for a list of such examples. The results of this paper show that we may obtain efficiently computable algorithms for tracking such structured classes of base experts. For example, (26) of Theorem 2 applies in this case, with $\rho_{\mathcal{E}}(n) = d\sqrt{2Kn\ln(d/K)} + dK\ln(d/K)$. The calculations of Section II-E may be easily modified for this case in a straightforward manner.

Example 3 (Tracking the best quantizers): The problem of limited-delay adaptive universal lossy source coding of individual sequences has recently been investigated in detail [21]–[23], [27], [33]–[35]. In the widely used model of fixed-rate lossy source coding at rate R, an infinite sequence of $[0,1]$-valued source symbols $x_1, x_2, \ldots$ is transformed into a sequence of channel symbols $y_1, y_2, \ldots$ which take values from the finite channel alphabet $\{1, 2, \ldots, M\}$, $M = 2^R$, and these channel symbols are then used to produce the ($[0,1]$-valued) reproduction sequence $\hat{x}_1, \hat{x}_2, \ldots$. The quality of the reproduction is measured by the average distortion $\sum_{t=1}^{n} d(x_t, \hat{x}_t)$, where d is some nonnegative bounded distortion measure. The squared error $d(x, x') = (x - x')^2$ is perhaps the most popular example.

The scheme is said to have overall delay at most δ if there exist nonnegative integers δ_1 and δ_2 with $\delta_1 + \delta_2 \leq \delta$ such that each channel symbol y_n depends only on the source symbols $x_1, \ldots, x_{n+\delta_1}$ and the reproduction $\hat{x}_n$ for the source symbol x_n depends only on the channel symbols $y_1, \ldots, y_{n+\delta_2}$. When $\delta = 0$, the scheme is said to have zero delay. In this case, y_n depends only on $x_1, \ldots, x_n$, and $\hat{x}_n$ on $y_1, \ldots, y_n$, so that the encoder produces y_n as soon as x_n becomes available, and the decoder can produce $\hat{x}_n$ when y_n is received. The natural reference class of codes (experts) in this case is the set of M-level scalar quantizers

$$\mathcal{Q} = \{Q : [0,1] \to \{c_1, \ldots, c_M\}, \{c_1, \ldots, c_M\} \subset [0,1]\} \ .$$

The relative loss with respect to the reference class $\mathcal{Q}$ is known in this context as the distortion redundancy. For the squared error distortion, the best randomized coding methods [23], [33], [35], with linear computational complexity with respect to the set $\mathcal{Q}$, yield a distortion redundancy of order $O(n^{-1/4}\sqrt{\log n})$.

The problem of competing with the best time-variant quantizer that can change the employed quantizer several times (i.e., tracking the best quantizer), was analyzed in [27], based on a combination of [23] and the tracking algorithm of [12]. There the best linear-complexity scheme achieves $O((C+1)\log n/n^6)$ distortion redundancy when an upper bound C on the number of switches in the reference class is known in advance. On the other hand, applying our scheme with $g = O(1)$ in the method of [27] and using the bounds in Section II-E, gives a linear-complexity algorithm with distortion redundancy $O((C+1)^{1/2}\log^{3/4}(n)/n^{1/4}) + O((C+1)/(\log^{1/2}(n)/n^{1/2}))$ if C is known in advance and only slightly worse $O((C+1)^{1/2}\log^{3/4}(n)/n^{1/4}) + O((C+1)\log(n)/n^{1/2})$ distortion redundancy if C is unknown. When $g = 2n^{\gamma} - 1$, the distortion redundancy for linear complexity becomes somewhat worse, proportional to $n^{-\frac{1}{2(2+\gamma)}}$ up to logarithmic factors.

V. Conclusion

We examined the problem of efficiently tracking large expert classes where the goal of the predictor is to perform as well as a given reference class. We considered prediction strategies that compete with the class of switching strategies that can segment a given sequence into several blocks, and follow the advice of a different base expert in each block. We derived a family of efficient tracking algorithms that, for any prediction algorithm $\mathcal{A}$ designed for the base class, can be implemented with time and space complexity $O(n^{\gamma}\log n)$ times larger than that of $\mathcal{A}$, where n is the time horizon and $\gamma \geq 0$ is a parameter of the algorithm. With $\mathcal{A}$ properly chosen, our algorithm achieves a regret bound of optimal order for $\gamma > 0$, and only $O(\log n)$ times larger than the optimal order for $\gamma = 0$ for all typical regret bound types we examined. For example, for predicting binary sequences with switching parameters, our method achieves the optimal $O(\log n)$ regret rate with time complexity $O(n^{1+\gamma}\log n)$ for any $\gamma \in (0,1)$.

While if an upper bound on the maximal number of switches in the reference class is known in advance and the base expert class is small, the optimal regret rate is achievable with an algorithm of linear computational complexity [12]. Our results show that the optimal rate is achievable with the slightly larger $O(n^{1+\gamma}\log n), \gamma > 0$, complexity even if the number of switches is not known in advance and the base expert class is large. It remains, however, an open question whether the optimal rate is achievable with a linear complexity algorithm in this case.

REFERENCES

[1] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning, and Games*. Cambridge: Cambridge University Press, 2006.

[2] A. V. Chernov and F. Zhdanov, “Prediction with expert advice under discounted loss,” in *ALT*, 2010, pp. 255–269.

[3] R. E. Krichevsky and V. K. Trofimov, “The performance of universal encoding,” *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 199–207, Mar. 1981.

[4] F. M. J. Willems, Y. N. Shtarkov, and T. J. Tjalkens, “The context-tree weighting method: Basic properties,” *IEEE Trans. Inform. Theory*, vol. IT-41, pp. 653–664, May 1995.

[5] E. Takimoto and M. K. Warmuth, “Path kernels and multiplicative updates,” *Journal of Machine Learning Research*, vol. 4, pp. 773–818, 2003.

[6] M. Herbster and M. K. Warmuth, “Tracking the best linear predictor,” *Journal of Machine Learning Research*, vol. 1, pp. 281–309, 2001.

[7] D. P. Helmbold and M. K. Warmuth, “Learning permutations with exponential weights,” *JMLR*, vol. 10, pp. 1705–1736, 2009.

[8] W. M. Koolen, M. K. Warmuth, and J. Kivinen, “Hedging structured concepts,” in *23rd Annual Conference on Learning Theory*, 2010.

[9] E. Hazan, A. Agarwal, and S. Kale, “Logarithmic regret algorithms for online convex optimization,” *Machine Learning Journal*, vol. 69, no. 2-3, pp. 169–192, 2007.

[10] F. M. J. Willems, “Coding for a binary independent piecewise-identically-distributed source,” *IEEE Trans. Inform. Theory*, vol. IT-42, pp. 2210–2217, Nov. 1996.

[11] G. I. Shamir and N. Merhav, “Low-complexity sequential lossless coding for piecewise-stationary memoryless sources,” *IEEE Trans. Inform. Theory*, vol. IT-45, pp. 1498–1519, July 1999.

[12] M. Herbster and M. K. Warmuth, “Tracking the best expert,” *Machine Learning*, vol. 32, no. 2, pp. 151–178, 1998.

[13] V. Vovk, “Derandomizing stochastic prediction strategies,” *Machine Learning*, vol. 35, no. 3, pp. 247–282, Jun. 1999.

[14] W. Koolen and S. de Rooij, “Combining expert advice efficiently,” in *Proceedings of the 21st Annual Conference on Learning Theory, COLT 2008*, Helsinki, Finland, July 2008, pp. 275–286.

[15] C. Monteleoni and T. S. Jaakkola, “Online learning of non-stationary sequences,” in *Advances in Neural Information Processing Systems 16*, S. Thrun, L. Saul, and B. Schölkopf, Eds. Cambridge, MA: MIT Press, 2004.

[16] S. de Rooij and T. van Erven, “Learning the switching rate by discretising Bernoulli sources online,” in *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics (AISTATS 2009)*, ser. JMLR Workshop and Conference Proceedings, vol. 5, Clearwater Beach, Florida USA, April 2009, pp. 432–439.

[17] F. Willems and M. Krom, “Live-and-die coding for binary piecewise i.i.d. sources,” in *Proceedings of the 1997 IEEE International Symposium on Information Theory (ISIT 1997)*, Ulm, Germany, June-July 1997, p. 68.

[18] A. György, T. Linder, and G. Lugosi, “Efficient tracking of the best of many experts,” in *Information and Communication Conference*, Budapest, Aug. 25–28 2008, pp. 3–4.

[19] E. Hazan and C. Seshadhri, “Efficient learning algorithms for changing environments,” in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 393–400.

[20] A. Kalai and S. Vempala, “Efficient algorithms for the online decision problem,” in *Proceedings of the 16th Annual Conference on Learning Theory and the 7th Kernel Workshop, COLT-Kernel 2003*, B. Schölkopf and M. Warmuth, Eds. New York, USA: Springer, Aug. 2003, pp. 26–40.

[21] T. Linder and G. Lugosi, “A zero-delay sequential scheme for lossy coding of individual sequences,” *IEEE Trans. Inform. Theory*, vol. 47, pp. 2533–2538, Sep. 2001.

[22] T. Weissman and N. Merhav, “On limited-delay lossy coding and filtering of individual sequences,” *IEEE Trans. Inform. Theory*, vol. 48, pp. 721–733, Mar. 2002.

[23] A. György, T. Linder, and G. Lugosi, “Efficient algorithms and minimax bounds for zero-delay lossy source coding,” *IEEE Transactions on Signal Processing*, vol. 52, pp. 2337–2347, Aug. 2004.

[24] S. Kozat and A. Singer, “Universal switching linear least squares prediction,” *IEEE Transactions on Signal Processing*, vol. 56, no. 1, pp. 189–204, Jan. 2008.

[25] ——, “Switching strategies for sequential decision problems with multiplicative loss with application to portfolios,” *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2192–2208, June 2009.

[26] ——, “Universal randomized switching,” *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1922–1927, March 2010.

[27] A. György, T. Linder, and G. Lugosi, “Tracking the best quantizer,” *IEEE Transactions on Information Theory*, vol. 54, pp. 1604–1625, Apr. 2008.

[28] E. Hazan and C. Seshadhri, “Adaptive algorithms for online decision problems,” *Elestronic Colloquium on Computational ComplexityProceedings of the 26th Annual International Conference on Machine Learning*, p. Report No. 88, 2007.

[29] T. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 2006.

[30] N. Merhav, “On the minimum description length principle for sources with piecewise constant parameters,” *IEEE Trans. Inform. Theory*, pp. 1962–1967, November 1993.

[31] N. Cesa-Bianchi and G. Lugosi, “Combinatorial bandits,” *Journal of Computer and System Sciences*, to appear.

[32] V. Dani, T. Hayes, and S. Kakade, “The price of bandit information for online optimization,” in *Proceedings of NIPS 2008.*, 2008.

[33] A. György, T. Linder, and G. Lugosi, “A ”follow the perturbed leader”-type algorithm for zero-delay quantization of individual sequences,” in *Proc. Data Compression Conference*, Snowbird, UT, USA, Mar. 2004, pp. 342–351.

[34] S. Matloub and T. Weissman, “Universal zero delay joint source-channel coding,” *IEEE Transactions on Information Theory*, vol. 52, pp. 5240–5250, 2006.

[35] A. György and G. Neu, “Near-optimal rates for limited-delay universal lossy source coding,” in *Proceedings of the IEEE International Symposium on Information Theory*, St. Petersburg, Russia, July-August 2011, pp. 2344–2348.